Exponential Bounds for DPLL Below the Satisfiability Threshold

Dimitris Achlioptas* Paul Beame† Michael Molloy‡

Abstract

For each $k \geq 4$, we give $r_k > 0$ such that a random k-CNF formula F with n variables and $\lfloor r_k n \rfloor$ clauses is satisfiable with high probability, but ORDERED-DLL takes exponential time on F with uniformly positive probability. Using results of [2], this can be strengthened to a high probability result for certain natural backtracking schemes and extended to many other DPLL algorithms.

1 Previous work

In the last twenty years a significant amount of work has been devoted to the study of randomly generated satisfiability instances and the performance of different algorithms on them. Historically, a major motivation for studying random instances has been the desire to understand the hardness of "typical" instances. Indeed, some of the better practical ideas in use today come from insights gained by studying the performance of algorithms on random k-SAT instances (defined below).

Let $C_k(n)$ denote the set of all possible disjunctions of k distinct, non-complementary literals (k-clauses) from some canonical set of n Boolean variables. A random k-CNF formula $F_k(n, m)$ is formed by selecting uniformly, independently, and with replacement m clauses from $C_k(n)$ and taking their conjunction. We will say that a sequence of random events $\mathcal{E}_n$ occurs with high probability (w.h.p.) if $\lim_{n\to\infty} \Pr[\mathcal{E}_n] = 1$ and with uniformly positive probability if $\liminf_{n\to\infty} \Pr[\mathcal{E}_n] > 0$.

It is widely believed that for each $k \geq 3$, there exists a constant c_k such that $F_k(n, m = cn)$ is w.h.p. satisfiable if $c < c_k$ and w.h.p. unsatisfiable if $c > c_k$. Currently, the best general bounds are $2^k \ln 2 - O(k) < c_k < 2^k \ln 2 - O(1)$, where by $c_k < c$ we mean that $F_k(n, cn)$ is w.h.p. unsatisfiable (analogously for $c_k > c$).

Let $res(F)$ denote the size of the minimal resolution refutation of a formula F (we define $res(F)$ to be infinite when F is satisfiable). A celebrated result of Chvátal and Szemerédi [5] asserts that for all $k \geq 3$ and every constant $c > 0$, w.h.p. $res(F_k(n, cn)) = 2^{\Omega(n)}$. Since $res(F)$ is always a lower bound on the time it takes a DPLL algorithm to determine that F is unsatisfiable, this implies that every such algorithm w.h.p. requires exponential time on $F_k(n, cn)$ when $c > c_k$.

In [2] we studied the behaviour of DPLL algorithms on random 3-CNF formulas when c is somewhat *below* the conjectured value for c_3 (≈ 4.2). Using standard techniques it is not hard to show that many natural DPLL algorithms when applied to such formulas, generate at least one unsatisfiable subproblem consisting of a random mixture of $m_2 = (1 - \epsilon)n$ 2-clauses and $m_3 = \Delta n$ 3-clauses. Our main contribution was the following theorem, implying that in that case the algorithm must spend an exponential amount of time before it can resolve such a subproblem and backtrack.

THEOREM 1.1. [2] *For every $\Delta, \epsilon > 0$, if F is the union of $(1 - \epsilon)n$ random 2-clauses and Δn random 3-clauses then w.h.p. $res(F) = 2^{\Omega(n)}$.*

Theorem 1.1 extends the result of Chvátal and Szemerédi [5] to accommodate random 2-clauses (such as those present in the subproblems generated by DPLL running on random k-CNF formulas). It asserts that such clauses do not significantly reduce the resolution complexity when $m_2 \leq (1 - \epsilon)n$ (for such m_2 w.h.p. the 2-clauses by themselves are satisfiable). Thus, we proved that certain natural DPLL algorithms require *exponential* time significantly below the generally accepted range for the random 3-SAT threshold. As an example, for ORDERED-DLL (which performs unit-clause propagation but, otherwise, sets variables in an a priori fixed random order/sign) we proved

THEOREM 1.2. [2] *For $c \geq 3.81$,* ORDERED-DLL *requires time $2^{\Omega(n)}$ on $F_3(n, cn)$ with uniformly positive probability.*

Indeed, we proved that Theorem 1.2 also holds for a few other (more intelligent) DPLL algorithms with 3.81 replaced by slightly larger values (but still significantly below 4.2). Moreover, we showed that for certain natural and effective forms of backtracking, "with uniformly positive probability" can be replaced with "w.h.p." in Theorem 1.2. Indeed, it appears that this should be true for *every* form of backtracking, but technical reasons make a general analysis rather unwieldy.

*Microsoft Research. Email: optas@microsoft.com.

†Computer Science & Engineering, Univ. of Washington. Email: beame@cs.washington.edu. Supported by NSF research grant CCR-0098066.

‡Computer Science, Univ. of Toronto. Email: molloy @cs.toronto.edu, Supported by NSERC and a Sloan Fellowship.

2 Our new contributions

The most obvious drawback of Theorem 1.2 is that it holds for values of c that are only *conjectured* (but not proven) to be in the "satisfiable regime". In this paper, we rectify this problem by showing that for all $k \geq 4$, the analogue to Theorem 1.2 holds for values of c that are *proven* to be in the satisfiable regime. Moreover, the gap between the density at which the algorithm begins to require exponential time, and the greatest density for which formulas are known to be satisfiable w.h.p. is large. In fact, the two densities are not even of the same asymptotic order in k. Specifically, we prove that

THEOREM 2.1. *With uniformly positive probability* ORDERED-DLL *requires time* $2^{\Omega(n)}$ *on* $F_k(n, cn)$ *if* $k = 4$ *and* $c \geq 7.5$ *or* $k \geq 5$ *and* $c \geq (11/k)\, 2^{k-2}$.

The values of c in Theorem 2.1 include values in the satisfiable regime by the following recent result of [3].

THEOREM 2.2. [3] $F_k(n, cn)$ *is w.h.p. satisfiable if* $k = 4$ *and* $c < 7.91$ *or* $k \geq 5$ *and* $c < 2^k \ln 2 - (k+4)/2$.

We note that Theorem 2.1 is, perhaps, only the simplest theorem that our techniques can deliver. ORDERED-DLL is an extension to a full DPLL algorithm of the non-backtracking UC algorithm [4]. We can, for example, prove analogues of Theorem 2.1 for any DPLL algorithm extending many other "myopic" [1] algorithms, that are much more sophisticated than UC. The arguments are technically more complex but in spirit they completely parallel the proof of Theorem 2.1.

3 The proof

Our strategy is to prove that, even for a satisfiable formula F, the algorithm will w.h.p. reach an unsatisfiable subformula F' where $res(F') = 2^{\Omega(n)}$. Thus, the algorithm will take $2^{\Omega(n)}$ steps just to determine that F' is unsatisfiable before backtracking away from it.

LEMMA 3.1. *Let F be a random CNF formula on n variables with $m_i \geq r_i n$ clauses of length i. If* $\lambda = \ln 2 + \sum_i r_i \ln(1-2^{-i}) < 0$ *then F is unsatisfiable w.h.p.*

Proof. If X is the expected number of satisfying assignments of F then $\mathbf{E}[X] = 2^n \prod_i \left(1-2^{-i}\right)^{m_i}$. Thus, $\Pr[X > 0] \leq \mathbf{E}[X] \leq \exp(\lambda n)$. □

We will say that a DPLL algorithm makes a "step" every time it assigns a value to a variable. Thus, the number of steps is non-decreasing (it stays constant as variables are unassigned during backtracking). Given an input formula F, we will denote the residual formula after t steps by F_t.

LEMMA 3.2. [4] *Let A be any DPLL algorithm extending* UC. *If during the first t steps of an execution of A on a random k-CNF formula no backtracking has occurred, then for each i, the set of i-clauses in F_t is uniformly random conditional on its size.*

DEFINITION 3.1. *For any fixed integers $k, i \geq 2$ and any real $c > 0$ define $f_i : [0,1] \to \mathbb{R}$ as*

$$f_i(x) = c\binom{k}{i} 2^{i-k}(1-x)^i x^{k-i} \quad .$$

LEMMA 3.3. [4, 1] *Assume that k, c, x_0 are such that $f_2(x) < 1 - x$ for all $x \in [0, x_0]$. Fix $x \in [0, x_0]$ and let $t = \lfloor xn \rfloor$. If we run any DPLL algorithm extending* UC *on a random k-CNF formula with $\lfloor cn \rfloor$ clauses then with uniformly positive probability all of the following hold after exactly t steps:*
a) No backtracking has occurred, b) No 0- or 1-clauses are present in F_t, c) For every $2 \leq i \leq k$, the number of i-clauses in F_t is $f_i(x) \cdot n + o(n)$.

Using Lemmata 3.1 and 3.3 we prove the following which readily implies Theorem 2.1.

LEMMA 3.4.

- *Let F be a random 4-CNF formula on n variables with $\lfloor cn \rfloor$ clauses, where $c \geq 7.5$. There exists t such that with uniformly positive probability, F_t is unsatisfiable and $res(F_t) = 2^{\Omega(n)}$.*
- *For fixed $k \geq 5$, let F be a random k-CNF formula on n variables with $\lfloor cn \rfloor$ clauses, $c \geq (11/k)\, 2^{k-2}$. There exists t such that with uniformly positive probability, F_t is unsatisfiable and $res(F_t) = 2^{\Omega(n)}$.*

Proof Elements. For $k = 4$, take $x_4 = 0.375$, $t = \lfloor x_4 n \rfloor$ and observe that for $c = 7.5$, $f_2(x) < (1-x)$ for all $x < x_0 \equiv 0.378$. To prove that F_t is unsatisfiable, let $r_i = 0.999 \times f_i(x_4)/(1-x_4)$ and apply Lemma 3.1. For $k \geq 5$, take $x_k = \frac{k-4}{k-1}$, $t = \lfloor x_k n \rfloor$ and argue analogously for $c = (11/k)\, 2^{k-2}$. For all k, the result for larger c follows by a monotonicity argument.

References

[1] D. Achlioptas. Lower bounds for random 3-SAT via differential equations. *TCS*, (1-2):159-185, 2001.

[2] D. Achlioptas, P. Beame, and M. Molloy. A sharp threshold in proof complexity. *STOC'01*, pp. 337–346.

[3] D. Achlioptas and Y. Peres. The random k-SAT threshold is $2^k \ln 2 - O(k)$. *STOC'03*, pp. 223–231.

[4] M.-T. Chao, J. Franco. Probabilistic analysis of a generalization of the unit-clause literal selection heuristics for the k-satisfiability problem. *Inform. Sci.*, 51(3):289–314, 1990.

[5] V. Chvátal, E. Szemerédi. Many hard examples for resolution. *JACM*, 35(4):759–768, 1988.